Attorney Docket No. BP 2483

TITLE: HIGH PERFORMANCE SCANNER FOR USER INPUT DEVICES

INVENTOR: Robert W. Hulvey

SPECIFICATION

BACKGROUND

1. Technical Field

The present invention relates generally to digital computers; and more particularly to wireless interfaces for allowing user input devices to communicate with digital computers.

2. Related Art

There are many user input devices for use with a digital computer, including standard keyboards, touchpads, mice and trackballs. Wireless communication technology has advanced rapidly over the past few years and there has been rapid development of wireless technologies for providing communication between input/output devices and their "host" computers. For example, wireless keyboards and mice now couple via wireless connections to their host computers. These "wireless" input devices are highly desirable since they do not require any hard-wired connections with their host computers. However, the lack of a wired connection also requires that the wireless input devices contain their own power supply, i.e., that they be battery powered.

In order to extend the life of its batteries, a wireless input devices often supports power saving modes of operation. For example, a wireless interface may include circuitry to provide for various levels of "power-down" modes to reduce power consumption when the device is inactive. When activity is detected, the interface circuitry will transition to a powered-up mode to facilitate communications between the user interface device and the computer and will then

return to a power-down mode after a predetermined interval of inactivity of the user interface device.

To obtain maximum power conservation it is important to minimize the amount of time that the interface circuitry must remain in a powered-up mode. Traditional methods of scanning input devices, such as keyboards, are comparatively inefficient and result in significantly reduced battery life for wireless input devices.

Thus, there is a need in the art for a method and apparatus for decreasing the amount of time to complete a scan of a user input device to thereby allow a wireless input device to operate for an extended period on a single battery life.

SUMMARY OF THE INVENTION

The present invention overcomes the shortcomings of the prior art by providing an improved wireless interface device that services communications between a wirelessly enabled host and at least one user input device. The wireless interface device is broadly comprised of a wireless interface unit that provides an interface between a user device and a wirelessly enabled host. The user input device comprises a switch matrix having a plurality of rows and columns. A keyboard scanning circuit in the wireless interface device is operable to scan the rows and columns of the user input device, wherein the scanning circuit detects operation of a key associated with the user device by detecting a transition in the voltage level of at least one row in the switch matrix from a first state to a second state and thereafter forces the row back to the first state thereby decreasing the scanning interval for detecting row transitions.

In a method of detecting inputs to a key switch matrix on a user input device in accordance with the present invention, control signals are applied to the rows and columns of the switch matrix

to place the rows and columns in a predetermined state. Activation of one of the keys in the switch matrix is detected based on a transition in the voltage level of at least one row in the switch matrix from a first state to a second state. The scanning logic of the user interface scans the state of the rows and columns in the key matrix in accordance with a predetermined scan sequence and the row is then forced back to the first state, thereby decreasing the scanning interval for detecting row transitions.

The reduced scanning interval provided by the method and apparatus of the present invention allows a wireless input device to operate with lower power consumption, thereby extending the operating period available using battery power.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1A is a system diagram illustrating a PC host and a wireless mouse that includes a wireless interface device constructed according to the present invention.

FIG. 1B is a system diagram illustrating a PC host and a wireless keyboard that includes a wireless interface device constructed according to the present invention.

FIG. 2 is a schematic block diagram illustrating the structure of a wireless mouse that includes a wireless interface device constructed according to the present invention.

FIG. 3 is a schematic block diagram illustrating the structure of a wireless keyboard that includes a wireless interface device constructed according to the present invention.

FIG. 4 is a block diagram illustrating a wireless interface device (integrated circuit) constructed according to the present invention.

FIG. 5 is a block diagram illustrating a wireless interface unit of the wireless interface device of FIG. 4.

FIG. 6 is a block diagram illustrating a processing unit of the wireless interface device of FIG. 4.

FIG. 7 is a block diagram illustrating an input/output unit of the wireless interface device of FIG. 4.

FIG. 8 is a block diagram generally showing the structure of an integrated circuit constructed according to the present invention with particular detail in the coupling of battery power to the units of the device.

FIG. 9 is a logic diagram illustrating operation according to the present invention.

FIG. 10 is a logic diagram illustrating operation according to the present invention in controlling the power consumption of a serviced device.

FIG. 11 is an illustration of the keyboard scan circuit components according to the present invention.

FIG. 12 is a timing diagram illustrating operation of the keyboard matrix circuitry operating in a first mode.

FIG. 13 is a flowchart illustration of the data processing steps carried out in accordance with the timing diagram of FIG. 12.

FIG. 14 is a timing diagram illustrating operation of the keyboard matrix circuitry operating in a second mode to identify a plurality of activated keys on a keyboard matrix.

DETAILED DESCRIPTION OF THE DRAWINGS

FIG. 1A is a system diagram illustrating a PC host 102 and a wireless mouse 104 that includes a wireless interface device constructed according to the present invention. As shown in FIG. 1A, the PC host 102 wirelessly couples to the wireless mouse 104. In the structure of FIG. 1A, the wireless mouse 104 includes a wireless interface device that operates to place the wireless mouse in any of a number of reduced power operating modes, including a power down mode in which battery life is substantially extended.

FIG. 1B is a system diagram illustrating a PC host 106 and a wireless keyboard 108 that includes a wireless interface device constructed according to the present invention. The wireless keyboard 108 is battery powered and operates for extended periods of time on a single set of batteries because of the greatly reduced power consumption operations according to the present invention.

FIG. 2 is a schematic block diagram illustrating the structure of a wireless mouse that includes a wireless interface device constructed according to the present invention. An integrated circuit 202 constructed according to the present invention serves as the wireless interface device and couples to various mouse inputs 210. These mouse inputs 210 include x-axis and y-axis inputs as well as a scroll input. The x-axis and y-axis inputs are often referred to a "quadrature" inputs. The components that produce the quadrature inputs are generally referred to at numeral 212 and may be constructed from optical inputs instead of from conventional mechanical inputs. Referenced via numeral 214 are the button inputs that are typical with a computer mouse and include the left button input, the middle/scroll button input, and the right button input. As is shown, each of the signals produced by the mouse are received by integrated circuit 202.

Integrated circuit 202 also couples to battery 204, crystal 206 that produces a 12 MHz reference frequency, EEPROM 208, and antenna 216. In one embodiment of the present invention, battery 204 comprises a pair of either AA batteries or AAA batteries. Antenna 216 is an internal antenna in the described because of the size constraints of the mouse and because of the relatively short distance between the PC host and the wireless mouse.

FIG. 3 is a schematic block diagram illustrating the structure of a wireless keyboard that includes a wireless interface device (integrated circuit 202) constructed according to the present invention. As shown in FIG. 3, integrated circuit 202 services a key scan matrix 202 that provides inputs from the keyboard. Indicators 304 include number, capitals, and scroll lights that are lit on the keyboard. The integrated circuit 202 couples to a battery 204, a crystal 206, an EEPROM 208, and an antenna 216.

In another embodiment (not shown in either FIG. 2 or FIG. 3), the integrated circuit 202 services both mouse and keyboard input and may reside internal to either the mouse or the keyboard. As will be apparent to those skilled in the art, multiplexing or signal sharing may be required, because the input signals differ. However, different signal lines may be dedicated for keyboard and for mouse inputs such that no signal sharing is required. As is apparent, when the integrated circuit 202 alone services both mouse and keyboard, input wired connectivity between the keyboard and the mouse is required.

FIG. 4 is a block diagram illustrating a wireless interface device (integrated circuit) constructed according to the present invention. As shown in FIG. 4, the wireless interface device 400 includes a processing unit 402, a wireless interface unit 404, an input/output unit 406, and a power management unit 408. The wireless interface unit 404 couples the wireless interface

device 400 to antenna 216. The wireless interface unit 404 can be adapted to operate according to the Bluetooth specification and in particular to the Human Interface Device (HID) portion of the Bluetooth specification. It will be understood by those skilled in the art, however, that the present invention can be adapted to work in conjunction with other wireless interface standards.

Processing unit 402, wireless interface unit 404, and input/output unit 406 couple with one another via a system on a chip (SOC) bus 410. Processing unit 402 includes a processing interface that may be used to couple the processing unit to one or more devices. Input/output unit 406 includes an input/output set of signal lines that couple the wireless interface device 400 to at least one user input device, such as a mouse or the keyboard incorporating the improved scan circuit described herein below.

FIG. 5 is a block diagram illustrating a wireless interface unit of the wireless interface device of FIG. 4. The wireless interface unit 404 includes a transmit/receive switch 502, a 2.4 GHz transceiver 504, a baseband core 506 which may be compatible with the Bluetooth standard, and a frequency synthesizer 508. Each of these components is generally known in the field and will be described in minimal detail herein.

The transmit/receive switch 502 couples to antenna 216 and switches between transmit and receive operations. The 2.4 GHz transceiver 504 performs all RF front-end operations and operates within a frequency band and on particular channels as are specified by the Bluetooth operating standard. The 2.4 GHz transceiver 504 couples to baseband core 506. Such coupling is performed via an RF control interface and an RF data interface. The RF control interface performs the necessary control operations to guarantee that the 2.4 GHz transceiver 504 and the baseband core 506 will operate consistently with desired operating specifications. The RF data

interface transfers both Rx and Tx data between the 2.4 GHz transceiver 504 and the baseband core 506. Frequency synthesizer 508 couples to the power management unit 408, to the external crystal 206 operating at 12 MHz, and to the 2.4 GHz transceiver 504. The frequency synthesizer 508 is controlled to provide an RF frequency for the 2.4 GHz transceiver 504 which is used to mix with the baseband signal received from the baseband core during a transmit operation and to mix with the received RF signal during a receive operation. The baseband core 506 couples to other wireless interface devices via the SOC bus 410.

FIG. 6 is a block diagram illustrating a processing unit 402 of the wireless interface device of FIG. 4. The processing unit 402 includes a microprocessor core 602, read only memory 606, random access memory 604, serial control interface 608, bus adapter unit 610, and multiplexer 612. The microprocessor core 602, ROM 606, RAM 604, serial control interface 608, bus adapter unit 610, and multiplexer 612 couple via a processor on a chip bus. Multiplexer 612 multiplexes an external memory interface between the processor on a chip bus and a test bus. The bus adapter unit 610 interfaces the processor on a chip bus with the SOC. The microprocessor core 602 includes a universal asynchronous receiver transmitter interface that allows direct access to the microprocessor core. Further, the serial control interface 608 provides a serial interface path to the processor on a chip bus.

FIG. 7 is a block diagram illustrating an input/output unit 406 of the wireless interface device of FIG. 4. The input/output unit 406 includes a keyboard scanning block 702, a mouse quadrature decoder block 704, and a GPIO control block 706. Each of the keyboard scanning block 702, the mouse quadrature decoder block 704, and the GPIO control block 706 couple to the SOC bus. Further, each of the keyboard scanning block 702, the mouse quadrature decoder

block 704, and the GPIO control block 706 couple to I/O via multiplexer 708. This I/O couples to the at least one user input device.

In another embodiment of the input/output unit 406, each of the keyboard scanning block 702, the mouse quadrature decoder block 704, and the GPIO control block 706 couples directly to external pins that couple to the at least one user input device.

FIG. 8 is a block diagram generally showing the structure of an integrated circuit constructed according to the present invention with particular detail in the coupling of battery power to the units of the device. Integrated circuit 800 of FIG. 8 includes a wireless interface unit 804, processing unit 802, input/output unit 806, and power management unit 808. The processing unit 802, wireless interface unit 804, and input/output unit 806 couple via a SOC bus 410. Further, as was previously described, input/output unit 806 couples to at least one user input device via I/O connection.

With the integrated circuit 800 of FIG. 8, a pad ring 814 surrounds a substantial portion of the components of the integrated circuit. The pad ring 814 couples directly to battery 204, which powers the pad ring. Further, input/output unit 806 and power management unit 808 couple directly to pad ring 814 to receive their power and voltage. However, processing unit 802 couples to pad ring 814 via processing unit voltage regulation circuitry 812. Further, the wireless interface unit 804 couples to pad ring 814 via wireless interface unit voltage regulation circuitry 810. The processing unit voltage regulation circuitry 812 is controlled by the power management unit 808 via control signal PU_EN. Further, the wireless interface unit voltage regulation circuitry 810 is controlled by the power management unit 808 using control signal WIU_EN.

The integrated circuit operates in four different power-conserving modes: (1) busy mode;

(2) idle mode; (3) suspend mode; and (4) power down mode. Busy mode, idle mode, and suspend mode are described in the Bluetooth specification. However, power down mode is unique to the present invention.

In busy mode, the Master (host computer) is actively polling the HID (wireless mouse, wireless keyboard, etc.) for data at a polling rate near 100 polls/second, or about once every 16 slot times. Continued user activity (keypad strokes, mouse motion, button presses, etc.) keeps the HID in busy mode. If there has been no activity for a few seconds (determined by particular settings), operation transitions to idle mode.

In idle mode, the HID requests the master (serviced host) to enter SNIFF mode with a SNIFF interval that is chosen based on desired latency and average power consumption. In one operation, the SNIFF interval is 50 ms, or about every 80 slot times. Although the HID can I/O Active immediately after an event, it may have to wait up to 100 mS to transmit its data to the host, and therefore must have enough buffer space to store 100 mS of events. If an event occurs, the HID requests the master to leave SNIFF mode. If there is no further activity for a longer period, the HID transitions from idle mode to suspend mode.

When entering suspend mode, there is a brief return on the connection state to busy mode to renegotiate the SNIFF interval to the suspend interval time. Then, the HID is parked. In suspend mode, a longer beacon interval can be used for a lower power state. When in suspend mode, any user input detected will result in the HID requesting to be unparked and transitioned back to the busy mode. When the HID is parked, it consumes less power than when the host is in SNIFF mode since the HID does not have to transmit. In suspend mode, the HID just listens to the beacons to remain synchronized to the master's frequency hopping clock. As long as the

master continues transmitting (meaning the host is not turned off) the HID will remain in suspend mode. If link loss occurs due to the host being turned off without warning, or the host moving out of range, the Lost Link state will be entered.

According to the present invention, the power down mode is also supported. In the power down mode, the power management unit 808 operates the processing unit voltage regulation circuitry 812 and the wireless interface unit voltage regulation circuitry 810 to power down the processing unit 802 and wireless interface unit 804, respectively. These states of operation will be described further with reference to FIGS. 9 and 10.

FIG. 9 is a logic diagram illustrating operation according to the present invention. As illustrated in FIG. 9, a wireless interface device operating according to the present invention operates in four separate power-conserving modes. These power conservation modes include the busy mode, the idle mode, the suspend mode and the power down mode. The state diagram of FIG. 9 shows how each of these modes is reached during normal operation.

When the wireless interface device is initially powered up, it enters the busy mode of operation. In the busy mode of operation, all features and wireless operations of the wireless interface device are enabled. As long as I/O activity continues, the wireless interface device remains in the busy mode. However, after expiration of a first timer with no I/O activity, the operation moves from the busy mode to the idle mode. Operation will remain in idle mode until the expiration of a second timer or until I/O activity occurs.

If while in the idle mode I/O activity occurs, operation returns to the busy mode. If in the idle mode, if timer 2 expires with no additional I/O activity, suspend mode is entered. While in suspend mode, if I/O activity occurs, operation returns to busy mode. However, if in suspend

mode, no additional I/O activity occurs until the expiration of a third timer, power down mode is entered. While in the power down mode, operation will remain in the power down mode until I/O activity occurs. When I/O activity occurs, operation of the wireless interface device will move from the power down mode to the busy mode.

FIG. 10 is a logic diagram illustrating operation according to the present invention in controlling the power consumption of a serviced device. As shown in FIG. 10, once operation in a particular power conservation state, e.g., busy mode, idle mode, suspend mode, and power down mode has commenced, operation will remain in that state until expiration of respective timer or I/O activity occurs (step 902).

When power conservation operation occurs to move from the busy mode to the idle mode (step 902), all portions of the wireless interface device remain powered (step 904). However, in the idle mode, the wireless interface unit enters a SNIFF mode in which some of its operations are reduced. Such operations were previously described with reference to FIG. 9. Further, additional information regarding this mode is available in the Bluetooth HID standard.

When the operation of the wireless interface device transitions from the idle mode to the suspend mode (step 908), all portions of the wireless interface device remain powered (step 910). However, the wireless interface unit of the wireless interface device enters the park mode, which consumes even less power than does the wireless interface unit when in the sniff mode.

When in the suspend mode, if an additional timer or inactivity period expires, the wireless interface device will transition to the power down mode (step 914). In the power down mode, the processing unit and wireless interface unit will be powered down (step 916). This power down operation will be performed in one embodiment by simply disconnecting a voltage source

from the processing unit in the wireless interface unit. One such technique for doing this is described with reference to FIG. 8. In the power down mode, the I/O unit will continue to be powered to allow it to sense the state of the user input device lines.

Finally, from any of the reduced power operating states, when I/O activity is sensed by the I/O block, the wireless input device will transition back to the busy mode (step 920). When such operation occurs, if the components have been powered down, they will be a powered up and will go through their boot operations (step 922). Then, in the busy mode, the wireless interface unit will operate in its normal state in which the master wireless device, i.e., wirelessly enabled host will poll the wireless interface device at 100 times per second. From each of steps 906, 912, 918, and 924, operation returns to step 902 wherein the current power conservation state will be kept until another event occurs.

Figure 11 is an illustration of a keyboard switch matrix 1102 connected to a key scan circuit 1104. The keyboard matrix 1102 comprises a plurality of columns 1106 and a plurality of rows 1108. In the embodiment shown in Figure 11, the plurality of columns 1108 comprises six columns C0-C5 and the plurality of rows comprises four rows, R0-R3. The embodiment illustrated in Figure 11 shows only a small portion of an actual keyboard matrix and it is understood by those skilled in the art that the number of rows and columns can be increased or decreased depending on the specific application.

A plurality of switches 1110 connect the respective rows and columns when a corresponding key is pressed by a user. In the illustration of Figure 11, switch 1110 connects row R0 and column C0 when the switch 1110 is pressed. Although a reference numeral has not been provided for each of the switches, it should be understood that a total of 24 switches 1110

are associated with the intersection of the rows and columns in Figure 11. For purposes of discussion, the twenty four illustrative switches 1110 in Figure 11 will be referred to as Switch 1, Switch 2, ..., Switch 24. When all of the respective switches in a particular row are open, the row will be pulled "high" by resistor 1112 that is connected to Vdd. Rows R0-R3 provide inputs to row decoder 1120 in the key scan circuit 1104, as will be discussed in greater detail below.

Key scan circuit 1104 comprises column/row control logic 1114 and driver logic 1115 that generate appropriate signals to control the state of the respective columns and rows. Driver logic 1115 comprises a tri-state driver 1116 and a driver 1118. The column/row control logic 1114 generates appropriate "high" and "low" signals that are provided to the inputs of the tri-state drivers 1116. The column/row control logic can change the state of a particular row or column by generating appropriate "enable" signals that control the operation of the tri-state drivers 1116 in the control logic 1115. For example, if the input of the tri-state driver 1116 is "high," the generation of an enable signal will cause the tri-state driver 1116 to apply the "high" signal at its output to drive the column or row "high." Conversely, if the input to the tri-state driver 1116 is "low," the generation of an enable signal will cause that tri-state driver to drive the column or row "low." The enable signals can be global enable signals intended to enable the tri-state drivers for all rows, e.g. ENB_R, or for all columns, e.g. ENB_C. The enable signals also can be directed to a tri-state driver for a particular row, e.g. ENB_R1, or for a particular column, e.g. ENB_C3.

The key scan circuit 1104 also comprises row decoder 1120 and column decoder 1122 that are operable to decode output signals received from the respective rows and columns in the keyboard matrix 1102. The decoded output signals from the row decoder 1120 and the column

decoder 1122 are provided to scan logic 1124 which generates a data stream indicating the state of various switches (keys) 1110.

The key scan circuit also comprises a switch transition detection circuit 1126 that receives output signals from the row decoder 1120 and the column decoder 1122. The switch transition detection circuit 1126 generates an "I/O Active" signal that is provided to the I/O block to cause the system to transition into the "busy" mode as described herein.

Operation of the keyboard scan circuitry can be understood by referring to the timing diagrams of Figures 12-14. Referring to Figure 12, the initial state of all of the rows and columns is analyzed beginning at the "Ready" reference line. The transitions to the left of the "Ready" reference are provided simply to clarify the "high" or "low" status of the rows and columns when processing begins. Beginning at the "Ready" reference point, ENB_C is low (active) and all columns are, therefore, driven low. All of the rows are driven high via the resistors 1112 shown in Figure 11. Also, at the "Ready" reference point, ENB_R is high indicating that the row tri-state driver is not driving the rows and all switches are open.

If, as an example, Key #9 is pressed, R0 transitions from "high" to "low." This transition is used as a trigger to latch (store) all row values. This transition also causes all ENB_Cs to transition from "low" to "high." Since all ENB_Cs are "high," the columns are no longer being driven "low" and, therefore, R0 transitions back to "high." The actual transition of R0 to "high" will be delayed somewhat by the RC constant combination of the line capacitance of column C2 plus row R0 and the resistor 1112. This slower-than-desired transition is shown by the dotted line under the top left-hand corner of the low-to-high transition. In the present invention, an ENB_R "low" pulse is initiated by the column/row control logic 1114 via the driver logic 1115

to speed up the low-to-high transition of R0.

Since switch #9 is still pressed, the column C2 will transition to "high." The "low" to "high" transition of column C2 is used as a trigger to latch all column values. After the column values have been latched, ENB_C transitions from "high" to "low" and column C2 transitions from "high" to "low." All other columns are also maintained in the "low" state by ENB_C being "low."

In the example shown in Figure 12, there is one high latched column value (C2) and one low latched row value (R0). The single latched column and the single latched row uniquely identify a single key switch (switch #9) and, therefore, there is no need to enter into a "scan" of other rows and columns. Thus, the scan signal remains "low" during the entire cycle.

The column/row control logic 1114, in conjunction with the driver logic 1115, is operable to generate all of the control signals necessary to control the state transitions described above. Furthermore, the switch transition detection circuit 1126 is operable to generate an "I/O Active" signal for the I/O block immediately upon receiving an output signal from the column decoder 1122 and/or the row decoder 1120 indicating that a switch has been activated. In this example, the "I/O Active" signal will be generated immediately by the switch transition detection circuit 1126 immediately upon detection of the transition of row R0 from "high" to "low" as a result of switch #9 being activated.

Figure 13 is a flowchart representation of the generalized signal processing steps to identify which key has been pressed. In step 1302, the system is in the idle state. ENB_C is set to "low;" all columns are driven "low" and all rows are driven "high." In step 1304, the system determines whether a key stroke has been detected (i.e., a row has transitioned to "low"). If no

key stroke is detected, processing returns to the idle state in step 1302. If a key stroke is detected in step 1304, however, the rows are latched in step 1306 and ENB_C is set to "high" in step 1308 to drive column C2 "high." In Step 1309, an ENB_R "low" pulse is initiated by the column/row control logic 114 via the driver logic 115 to speed up the low-to-high transition of R0. In step 1310 the columns are "latched" and in step 1312 the columns are released. In step 1314, the system determines whether the number of columns is equal to 1 or whether the number of rows is equal to 1. If either of these conditions is met, processing proceeds to step 1316 indicating that no scanning is needed because the keys that were pressed have been uniquely identified and these keys are reported in step 1320. If, however, the test in step 1314 indicates that the number of columns is greater than 1 and the number of rows is greater than 1, processing proceeds to step 1318 where the columns are scanned as discussed in greater detail below. After the scanning process has been completed, the key numbers are reported in step 1320 and processing returns to the idle state in step 1302.

Operation of the scan mode can be understood by referring to the timing diagram of Figure 14. The initial state of all of the rows and columns is analyzed beginning at the "Ready" reference line. Again, the transitions to the left of the "Ready" reference are provided simply to clarify the "high" or "low" status of the rows and columns when processing begins. Beginning at the "Ready" reference point ENB_C is low (active) and all columns are, therefore, driven low. All of the rows are driven high via the resistors 1112 shown in Figure 11.

If, as an example, key #1, key #2, and key #5 are pressed, R0 and R1 transition from "high" to "low." These transitions are used as a trigger to latch (store) all row values. This transition also causes ENB_C to transition from "low" to "high." In addition ENB_C0, ENB_C1

and ENB_C2-5 will also transition from "low" to "high." Since ENB_C is "high," the columns are no longer being driven "low" and, therefore, R0 and R1 transition back to "high." The actual transition of R0 and R1 to "high" will be delayed somewhat by the RC constant combination of the line capacitance of columns C0 and C1 and the resistors 1112. These slower-than-desired transitions are shown by the dotted lines under the top right-hand corner of the low-to-high transitions for rows R0 and R1. In the present invention, an ENB_R "low" pulse is initiated by the column/row control logic 1114 via the driver logic 1115 to speed up the low-to-high transition of R0 and R1.

Since switch #1, switch #2 and switch #5 are still pressed, the columns C0 and C1 will transition to "high." The "low" to "high" transition of columns C0 and C1 are used as a trigger to latch all column values. Two columns, C0 and C1 are latched in the high state.

Because the column decoder 1122 and the row decoder 1120 determine that more than one column has been latched, and more than one row has been latched, the system enters scan mode, the SCAN signal goes "high," and the scan begins for the two columns, C0 and C1, latched in the "high" condition. First, ENB_C0 is driven "low" and, therefore, column C0 is driven low. With C0 driven low, the rows corresponding to activated switches will be driven low because the activated switch connects those rows to C0. In this example, switch #1 causes R0 to be driven low and switch #2 causes R1 to be driven low. The transition of rows R0 and R1 is detected by the row decoder 1120 and the scan logic 1126 to indicate that the switches corresponding to C0 and rows R0 and R1 are activated, thereby identifying that switch #1 and switch #2 are activated. ENB_C0 then makes a transition from "low" to "high" and, therefore, C0 is no longer driven "low." Rows R0 and R1, therefore, transition from "low" to "high,"

although the actual transition of R0 and R1 to "high" will be delayed somewhat by the RC constant combination of the line capacitance of columns C0, C1 and the resistors 1112. Once again, these slower-than-desired transitions are shown by the dotted lines under the top right-hand corner of the low-to-high transitions for rows R0 and R1. An ENB_R "low" pulse is initiated by the column/row control logic 1114 via the driver logic 1115 to speed up the low-to-high transition of R0 and R1.

Scanning continues with ENB_C1 transitioning from "high" to "low" thus driving C1 "low." With C1 driven low, the rows corresponding to activated switches will be driven low because the activated switch connects those rows to C1. In this example, switch #5 causes R0 to be driven low. The transition of row R0 is detected by the row decoder 1120 and the scan logic 1126 to indicate that the switch corresponding to C1 and row R0 is activated, thereby identifying that switch #5 is activated. ENB_C1 then makes a transition from "low" to "high" and, therefore, C1 is no longer driven "low." Row R0, therefore, transitions from "low" to "high." The actual transition of R0 to "high" will be delayed somewhat by the RC constant combination of the line capacitance of column C1 and the resistor 1112. Once again, this slower-than-desired transition is shown by the dotted line under the top right-hand corner of the low-to-high transition for row R0. An ENB_R "low" pulse is initiated by the column/row control logic 1114 via the driver logic 1115 to speed up the low-to-high transition of R0. With the scan of the columns completed, the scan logic will have reported switch #1, switch #2 and switch #5 as active. The system then returns to the "Ready" state.

The invention disclosed herein is susceptible to various modifications and alternative forms. Specific embodiments, therefore, have been shown by way of example in the drawings

and detailed description. It should be understood, however, that the drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the invention is to cover all modifications, equivalents and alternatives falling within the spirit and scope of the present invention as defined by the claims.